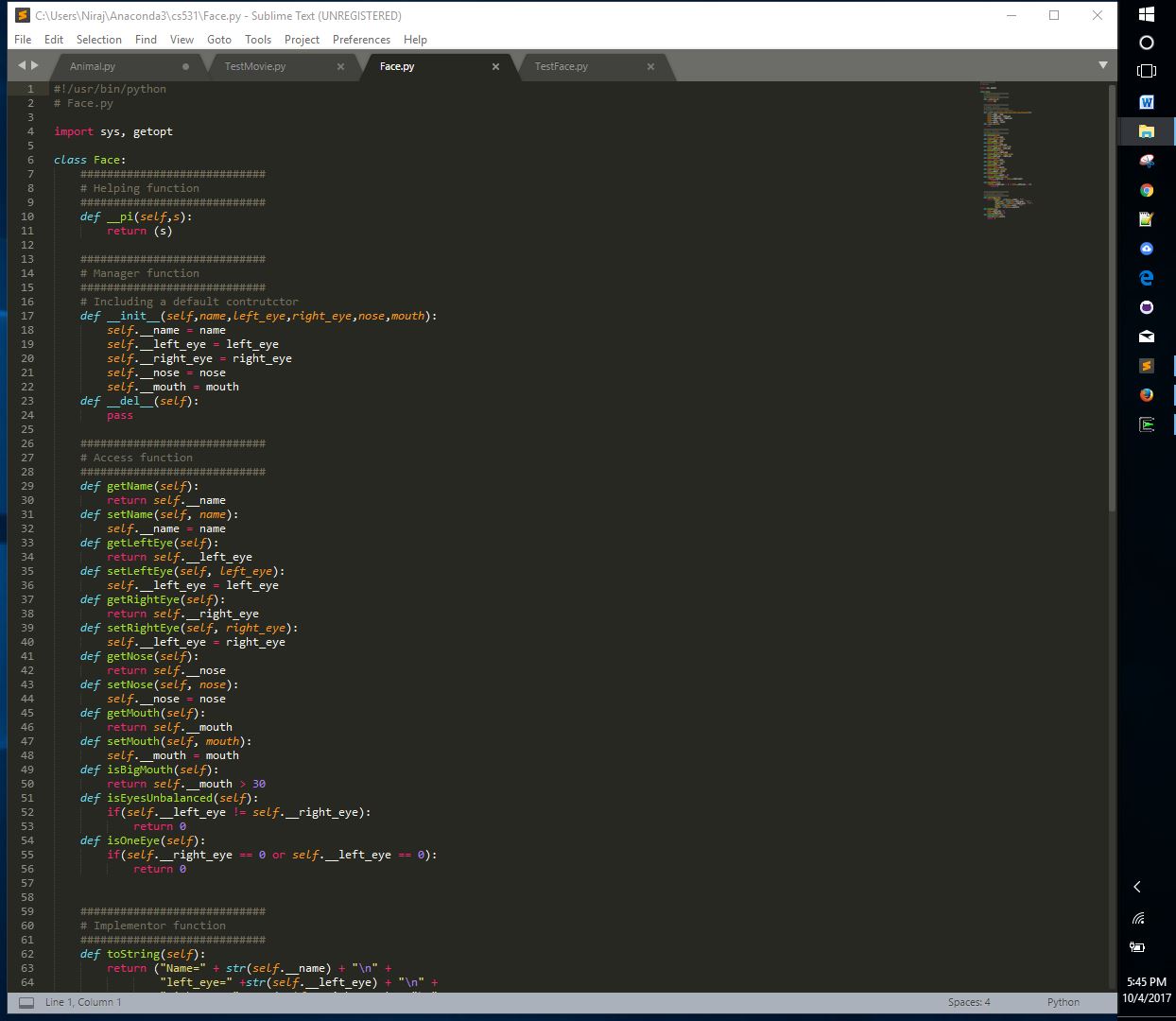
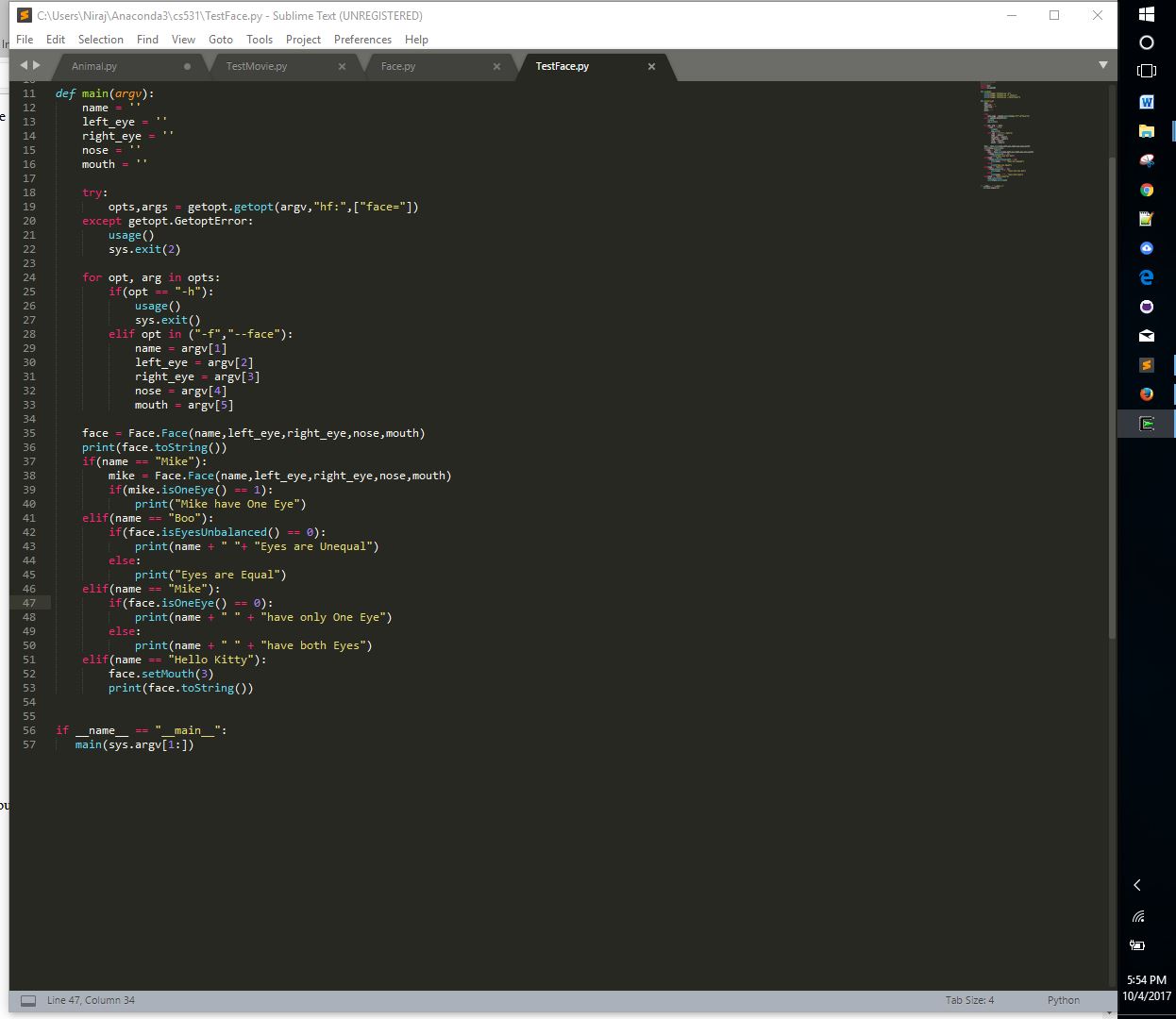
**Name : Niraj Thanki SID : 19376 CLASS : CS531**

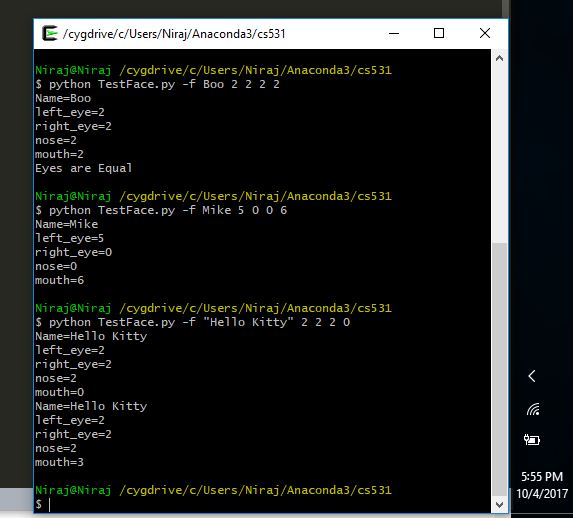
**Face.py**



**TestFace.py**



**OutPut :**

****

Source Code :

**Face.py**

#!/usr/bin/python

# Face.py

import sys, getopt

class Face:

############################

# Helping function

############################

def \_\_pi(self,s):

return (s)

############################

# Manager function

############################

# Including a default contrutctor

def \_\_init\_\_(self,name,left\_eye,right\_eye,nose,mouth):

self.\_\_name = name

self.\_\_left\_eye = left\_eye

self.\_\_right\_eye = right\_eye

self.\_\_nose = nose

self.\_\_mouth = mouth

def \_\_del\_\_(self):

pass

############################

# Access function

############################

def getName(self):

return self.\_\_name

def setName(self, name):

self.\_\_name = name

def getLeftEye(self):

return self.\_\_left\_eye

def setLeftEye(self, left\_eye):

self.\_\_left\_eye = left\_eye

def getRightEye(self):

return self.\_\_right\_eye

def setRightEye(self, right\_eye):

self.\_\_left\_eye = right\_eye

def getNose(self):

return self.\_\_nose

def setNose(self, nose):

self.\_\_nose = nose

def getMouth(self):

return self.\_\_mouth

def setMouth(self, mouth):

self.\_\_mouth = mouth

def isBigMouth(self):

return self.\_\_mouth > 30

def isEyesUnbalanced(self):

if(self.\_\_left\_eye != self.\_\_right\_eye):

return 0

def isOneEye(self):

if(self.\_\_right\_eye == 0 or self.\_\_left\_eye == 0):

return 0

############################

# Implementor function

############################

def toString(self):

return ("Name=" + str(self.\_\_name) + "\n" +

"left\_eye=" +str(self.\_\_left\_eye) + "\n" +

"right\_eye=" +str(self.\_\_right\_eye) + "\n" +

"nose=" +str(self.\_\_nose) + "\n" +

"mouth=" +str(self.\_\_mouth))

def blind(self):

self.\_\_left\_eye = 0

self.\_\_right\_eye = 0

def removeMouth(self):

remove(self.\_\_mouth)

return 0

**TestFace.py**

#!/usr/bin/python

#TestFace.py

import Face

import sys,getopt

def usage():

print("Usage: TestFace.py -h")

print("Usage: TestFace.py -f <face>")

print("Usage: TestFace.py --face=<face>")

def main(argv):

name = ''

left\_eye = ''

right\_eye = ''

nose = ''

mouth = ''

try:

opts,args = getopt.getopt(argv,"hf:",["face="])

except getopt.GetoptError:

usage()

sys.exit(2)

for opt, arg in opts:

if(opt == "-h"):

usage()

sys.exit()

elif opt in ("-f","--face"):

name = argv[1]

left\_eye = argv[2]

right\_eye = argv[3]

nose = argv[4]

mouth = argv[5]

face = Face.Face(name,left\_eye,right\_eye,nose,mouth)

print(face.toString())

if(name == "Mike"):

mike = Face.Face(name,left\_eye,right\_eye,nose,mouth)

if(mike.isOneEye() == 1):

print("Mike have One Eye")

elif(name == "Boo"):

if(face.isEyesUnbalanced() == 0):

print(name + " "+ "Eyes are Unequal")

else:

print("Eyes are Equal")

elif(name == "Mike"):

if(face.isOneEye() == 0):

print(name + " " + "have only One Eye")

else:

print(name + " " + "have both Eyes")

elif(name == "Hello Kitty"):

face.setMouth(3)

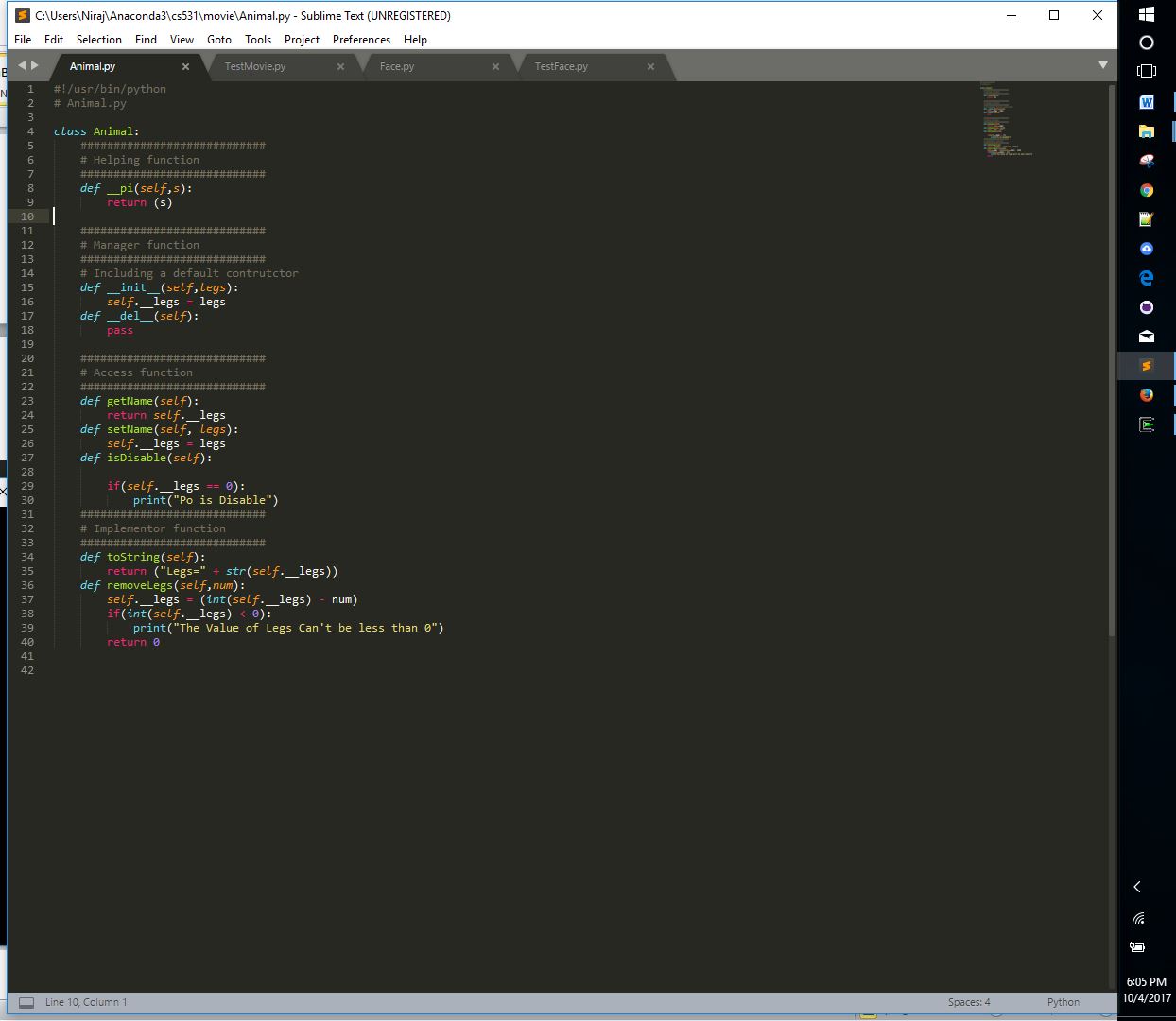
print(face.toString())

if \_\_name\_\_ == "\_\_main\_\_":

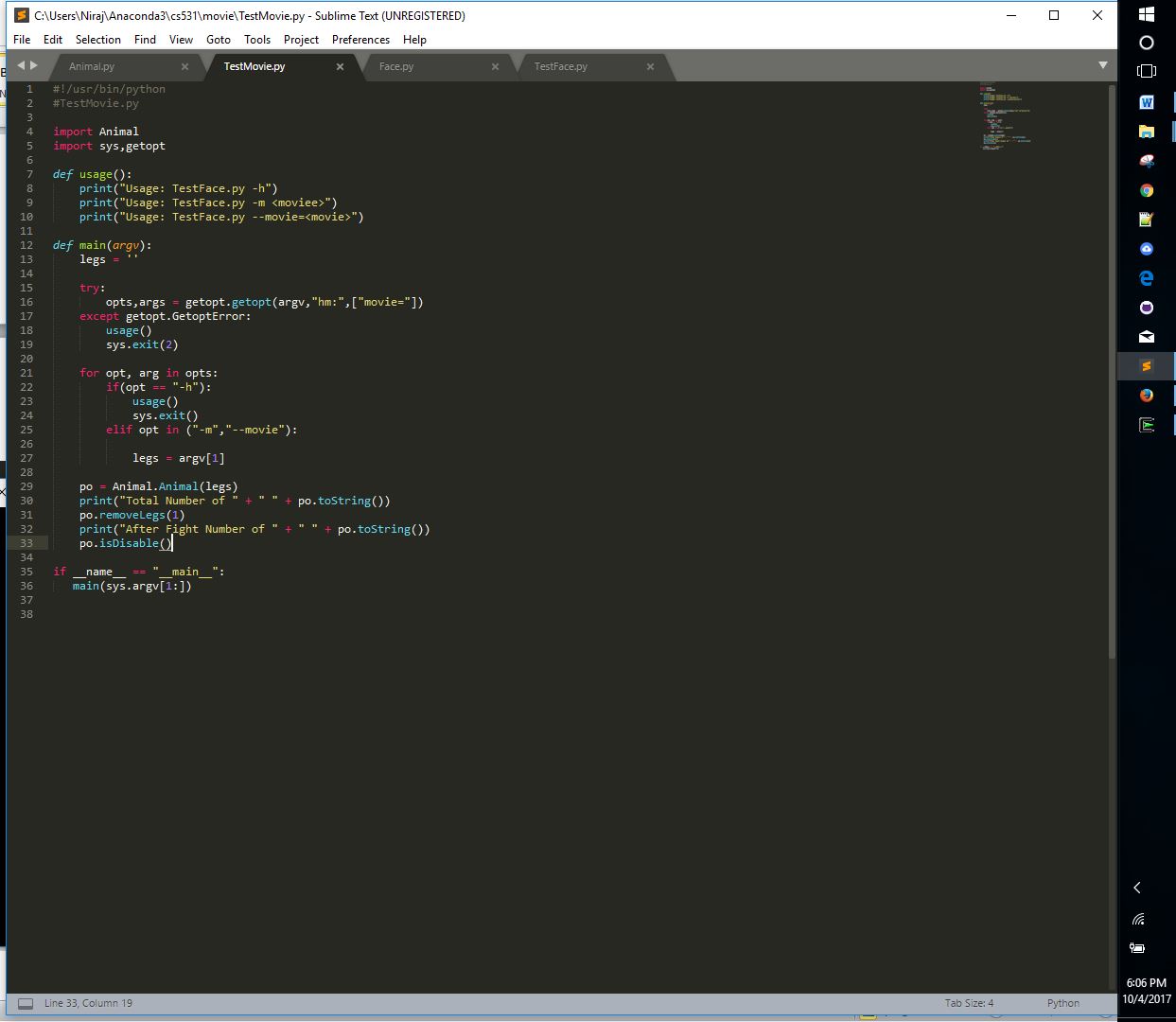
main(sys.argv[1:])

**Q-41**

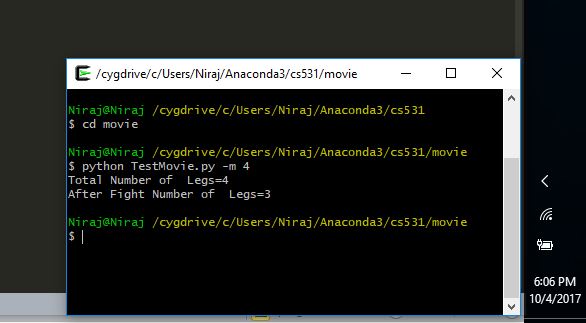
**Animal.py**

****

**TestMovie.py**

****

**Output :**

****

**Source Code :**

**Animal.py**

#!/usr/bin/python

# Animal.py

class Animal:

############################

# Helping function

############################

def \_\_pi(self,s):

return (s)

############################

# Manager function

############################

# Including a default contrutctor

def \_\_init\_\_(self,legs):

self.\_\_legs = legs

def \_\_del\_\_(self):

pass

############################

# Access function

############################

def getName(self):

return self.\_\_legs

def setName(self, legs):

self.\_\_legs = legs

def isDisable(self):

if(self.\_\_legs == 0):

print("Po is Disable")

############################

# Implementor function

############################

def toString(self):

return ("Legs=" + str(self.\_\_legs))

def removeLegs(self,num):

self.\_\_legs = (int(self.\_\_legs) - num)

if(int(self.\_\_legs) < 0):

print("The Value of Legs Can't be less than 0")

return 0

**TestMovie.py**

#!/usr/bin/python

#TestMovie.py

import Animal

import sys,getopt

def usage():

print("Usage: TestFace.py -h")

print("Usage: TestFace.py -m <moviee>")

print("Usage: TestFace.py --movie=<movie>")

def main(argv):

legs = ''

try:

opts,args = getopt.getopt(argv,"hm:",["movie="])

except getopt.GetoptError:

usage()

sys.exit(2)

for opt, arg in opts:

if(opt == "-h"):

usage()

sys.exit()

elif opt in ("-m","--movie"):

legs = argv[1]

po = Animal.Animal(legs)

print("Total Number of " + " " + po.toString())

po.removeLegs(1)

print("After Fight Number of " + " " + po.toString())

po.isDisable()

if \_\_name\_\_ == "\_\_main\_\_":

main(sys.argv[1:])

**TestMovie.py**

#!/usr/bin/python

#TestMovie.py

import Animal

import sys,getopt

def usage():

print("Usage: TestFace.py -h")

print("Usage: TestFace.py -m <moviee>")

print("Usage: TestFace.py --movie=<movie>")

def main(argv):

legs = ''

try:

opts,args = getopt.getopt(argv,"hm:",["movie="])

except getopt.GetoptError:

usage()

sys.exit(2)

for opt, arg in opts:

if(opt == "-h"):

usage()

sys.exit()

elif opt in ("-m","--movie"):

legs = argv[1]

po = Animal.Animal(legs)

print("Total Number of " + " " + po.toString())

po.removeLegs(1)

print("After Fight Number of " + " " + po.toString())

po.isDisable()

if \_\_name\_\_ == "\_\_main\_\_":

main(sys.argv[1:])